대 한 민 국 특 허 청
KOREAN INDUSTRIAL PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Industrial Property Office.

출 원 번 호 :
Application Number — 특허출원 1999년 제 23092 호

출 원 년 월 일 :
Date of Application — 1999년 06월 19일

출 원 인 :
Applicant(s) — 한국과학기술원

CERTIFIED COPY OF PRIORITY DOCUMENT

2000 년 03 월 21 일

특 허 청
COMMISSIONER

【서류명】 출원서
【권리구분】 특허
【수신처】 특허청장
【참조번호】 2
【제출일자】 1999.06.19
【발명의 명칭】 3 차원 그래픽 텍스쳐 맵핑용 캐쉬 메모리 및 그의 캐쉬 미스페 널티 저감방법
【발명의 영문명칭】 Cache memory for 3D graphic texture and its cache misspena lty reducing method
【출원인】
【명칭】 한국과학기술원
【출원인코드】 3-1998-098866-1
【대리인】
【성명】 이원희
【대리인코드】 9-1998-000385-9
【포괄위임등록번호】 1999-004558-0
【발명자】
【성명의 국문표기】 박세정
【성명의 영문표기】 PARK,Se Jeong
【주민등록번호】 720101-1822218
【우편번호】 305-701
【주소】 대전광역시 유성구 구성동 373-1,한국과학기술원 전기전자과
【국적】 KR
【발명자】
【성명의 국문표기】 유회준
【성명의 영문표기】 YOO,Hoi Jun
【주민등록번호】 600730-1025311
【우편번호】 305-340
【주소】 대전광역시 유성구 도룡동 현대아파트 103동 705호
【국적】 KR
【발명자】
【성명의 국문표기】 박규호
【성명의 영문표기】 PARK,Kyu Ho

【주민등록번호】 501019-1030510

【우편번호】 305-338

【주소】 대전광역시 유성구 구성동 한빛아파트 132동 202호

【국적】 KR

【심사청구】 청구

【취지】 특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사 를 청구합니다. 대리인
이원희 (인)

【수수료】

【기본출원료】	20 면	29,000	원
【가산출원료】	3 면	3,000	원
【우선권주장료】	0 건	0	원
【심사청구료】	6 항	301,000	원
【합계】	333,000 원		

【요약서】

【요약】

본 발명은 PC용 고성능 3차원 그래픽 카드, 3차원 게임기 및 기타 소형 고성능 3차원 그래픽을 요구하는 분야에 적용 가능한 텍스쳐 맵핑용 캐쉬 메모리에 관한 것으로서, 특히 3차원 그래픽 시스템에서 트라이리니어 인터폴레이션(trilinear interpolation)을 이용하는 밉맵핑(mipmapping)에 의한 텍스쳐 맵핑 가속을 위하여, 적당한 크기의 워킹 셋만큼의 텍스쳐만을 저장하며, 한 클럭 사이클만에 트라이리니어 인터폴레이션을 수행하기 위하여 필요한 8개의 텍셀을 모두 엑세스하여 최종 텍셀의 값을 구해 낼 수 있는 특수한 구조를 갖는 캐쉬 메모리와 앞으로 필요하게 될 텍스쳐들을 하드웨어적으로 예측하여 프리페칭(hardware-predicted prefetching) 하므로써 캐쉬 미스(cache miss) 시에 발생하는 페널티(penalty)를 줄일 수 있는 캐쉬 미스 페널티 저감방법에 관한 것이다. 본 발명의 캐쉬 메모리 및 캐쉬 미스 페널티 저감방법은 기존의 하드웨어적인 텍스쳐 맵핑 가속화 방법보다 다양한 크기의 텍스쳐 이미지들에 대해서 효율적으로 적용이 가능하면서, 소형 저가의 시스템에서도 하드웨어적인 텍스쳐 맵핑 가속을 가능하게 한다.

【대표도】

도 7

【명세서】

【발명의 명칭】

3차원 그래픽 텍스쳐 맵핑용 캐쉬 메모리 및 그의 캐쉬 미스 페널티 저감방법{Cache memory for 3D graphic texture and its cache miss penalty reducing method}

【도면의 간단한 설명】

도 1은 밉매핑(mipmapping)의 개념도

도 2는 트라이리니어 인터폴레이션(trilinear interpolation)의 예제도

도 3은 텍스램(texram)의 구조도

도 4는 텍스램에서의 텍셀(texel) 배치도

도 5는 클립맵(clipmap)의 개념도

도 6은 본 발명의 일 실시예에 따른 텍스쳐 맵핑용 캐쉬 메모리의 개념도

도 7은 본 발명의 일 실시예에 따른 텍스쳐 맵핑용 캐쉬 메모리의 구조도

도 8은 텍스쳐 서브클립(subclip)의 예측 및 교체 과정을 나타낸 도

도 9는 텍스쳐 스택층(texture stack layer)의 예측 및 교체 과정을 나타낸 도

* 도면의 주요부분에 대한 부호의 설명 *

10 : 제1 DRAM 뱅크 20 : 제2 DRAM 뱅크

21 : CAM(Content Addressable Memory)

30 : 서브클립 로더 40 : 트라이리니어 인터폴레이터

50 : 서브클립 예측기 60 : 제어기

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<15> 본 발명은 PC용 고성능 3차원 그래픽 카드, 3차원 게임기 및 기타 소형 고성능 3차원 그래픽을 요구하는 분야에 적용 가능한 텍스쳐 맵핑용 캐쉬 메모리에 관한 것으로서, 상세하게는 트라이리니어 인터폴레이션을 이용한 밉맵핑에 의하여 텍스쳐 맵핑을 가속화시킬 수 있는 3차원 그래픽 텍스쳐 맵핑용 캐쉬 메모리 및 앞으로 필요하게 될 텍스쳐들을 하드웨어적으로 예측하여 프리페칭(hardware- predicted prefetching)하므로써 캐쉬 미스 시에 발생하는 페널티를 줄일 수 있는 캐쉬 미스 페널티 저감방법에 관한 것이다.

<16> 3차원 그래픽 시스템에서 보다 현실감 있는 장면을 얻기 위한 방법으로 텍스쳐 맵핑(texture mapping) 기술이 주로 이용된다. 텍스쳐 맵핑이란, 3차원 물체의 표면에 질감을 주기 위해서 실제 카메라로부터 촬영된 2차원 이미지를 물체 표면에 입히는 것을 말하며, 텍스쳐(texture)란 이러한 2차원 이미지를 말하고, 텍스쳐 내의 각 점들은 스크린 상의 픽셀(pixel)에 대비해서 텍셀(texel)이라 부른다.

<17> 그러나, 이 기술을 적용함에 있어서, 계산량을 줄이기 위하여 2차원 이미지, 즉 텍스쳐를 3차원 물체의 표면 위에 입힌 후, 2차원의 스크린 위로 투영시키는 것이 아니고(

스크린에 보이지 않는 부분까지 2차원 텍스쳐를 입히는 계산을 해야 하므로), 일반적으로 2차원 스크린에 투영된 이미지의 각 픽셀로부터 이들의 3차원 공간상의 물체 표면의 좌표를 구한 후, 이에 해당하는 2차원 텍스쳐 좌표를 계산해서 해당되는 텍셀을 찾아서 스크린 상의 픽셀의 색을 정하게 된다.

<18> 이러한 텍스쳐 맵핑 기술은 스크린 상에 표시되는 한 픽셀에 맵핑 되는 텍스쳐 공간상의 영역이 정확히 한 텍셀 영역으로 맵핑 되지 않기 때문에 화면상의 에이리어징 현상을 일으키게 된다.

<19> 밉매핑(mipmapping)의 개념을 나타낸 도 1을 예로 들면, 도 1의 우측의 2차원 스크린 상으로 투영된 도로와 같은 경우, 도로의 a나 b 영역에 해당하는 관측자에 가까운 픽셀들은 텍스쳐 공간상에 하나의 텍셀로 맵핑 된다고 가정하면, c나 d와 같이 관측자로부터 먼 영역의 픽셀은 텍스쳐 공간상에서 여러 개의 텍셀 영역으로 맵핑 되므로, 이 때 여러 개의 텍셀 값을 대표할 수 있는 텍스쳐 값을 가져와서 스크린상의 한 픽셀 값으로 맵핑 식혀 주지 않으면 영상이 찌그러지는 결과를 가져오게 된다.

<20> 이러한 여러 개의 텍셀들의 대표되는 값을 구하는 쉬운 방법으로, 텍스쳐 공간으로 맵핑된 영역 내의 모든 텍셀 값의 평균을 구해서 이용하는 방법이 많이 사용되고 있으며, 이를 텍스쳐 필터링(texture filtering)이라고 한다.

<21> 상술한 텍스쳐 필터링은 많은 시간을 요구하므로, 이를 좀 더 빠르게 하기 위한 방법으로 많이 사용되는 방법이 밉맵핑에 의한 방법이다.

<22> 밉맵핑은 도 1의 왼쪽에서와 같이 원래 텍스쳐 이미지(도면상에서 LOD 0에서의 이

미지가 원래 텍스쳐 이미지다. LOD(Level Of Detail)란 화면상의 한 픽셀이 텍스쳐 공간상의 몇 개의 텍셀 영역을 커버하는지를 나타내는 값이다.)를 미리 필터링 및 서브샘플링(subsampling)해서 여러 LOD 레벨의 텍스쳐들(mipmap)을 만들어 두고, 현재 화면상에 뿌려지는 픽셀이 텍스쳐 공간으로 맵핑될 때 계산되어지는 LOD 및 (u, v) 좌표 값에 의해서 밉맵핑상에서 적절한 LOD 레벨들로부터 필요로 하는 텍셀을 가져와서 화면상에 뿌리는 방식이다.

<23> 여기서, 텍스쳐 공간으로 맵핑된 LOD 및 (u, v) 값은 소수점 이하의 값을 가지고(스크린 상의 정수치인 (x, y) 좌표가 변환 매트릭스(transform matrix)를 거쳐서 나온 값이므로 소수점 이하의 값을 가지게 된다), 현재 밉맵상에 존재하는 값들은 LOD 및 (u, v)가 정수인 위치에서의 값들이다. 그러므로 스크린 상에서 현재 그리고자하는 픽셀이 텍스쳐 공간으로 맵핑된 정확한 LOD 및 (u, v) 위치에서의 텍스쳐 값은 현실적으로 존재하지 않는다. 그러나 밉맵 상에서 이 값과 가장 가까운 정수치 위치에서의 값들을 읽고 이들을 정확한 위치에서의 거리 차와의 관계를 이용해서 보간하게 된다.

<24> 예를 들어, 도 2에서와 같이 LOD=0.6, u=23.25, v=30.50으로 맵핑된 픽셀에 대한 텍셀 값을 구하는 방법은 아래와 같다.

<25> 1) LOD가 0.6 이므로, 먼저 LOD 0에서 u=23.25, v=30.50에 가장 가까운 정수 좌표 위치((23, 30),(23, 31),(24, 30),(24, 31))에 있는 텍셀 값들을 읽어서 실제 정확한 좌표 위치와 이들과의 거리 관계를 이용해서 LOD 0에서의 대표되는 값을 구한다.

<26> 2) LOD 1에서 대표되는 값을 구하기 위해서 u=23.25/2=11.62, v=30.50/2=15.25 에서 가장 가까운 정수 위치((11, 15), (11, 16), (12, 15), (12, 16))에서의 텍셀 값들을 읽어서 실제 정확한 좌표 위치와 이들과의 거리 관계를 이용해서 LOD 1에서 대표되는 값을 구한다.

<27> 3) LOD 0과 LOD 1에서 대표되는 두 값을 실제 정확한 LOD 값(0.6) 과의 차이를 이용해서 도 2와 같이 최종적인 대표 텍셀 값을 구하게 된다.

<28> 이와 같이 밉맵핑 방식은 빠른 텍스쳐 맵핑을 위해서 미리 여러 LOD 레벨의 텍스쳐들을 만들어 두고, 프로그램 수행시에 대표되는 텍셀의 값을 구하기 위해 8개의 텍셀을 읽고, 인터폴레이션(이러한 인터폴레이션 방식을 트라이리니어 인터폴레이션이라 한다)을 행한 후 이를 대표 값으로 이용하게 된다.

<29> 이는 정확한 필터링에 의한 방법보다는 훨씬 빠른 방법이지만, 여전히 실시간 3 차원 그래픽에서 밉맵핑을 위해 8개의 텍셀을 메모리로부터 순차적으로 읽는데 걸리는 시간은 큰 문제점이 되고 있다.

<30> 따라서, 이러한 문제점을 해결하기 위한 고속 텍스쳐 맵핑을 위한 방법으로 도 3에서와 같이 두 개의 LOD 레벨의 텍셀들을 저장할 수 있는 메모리 뱅크(memory bank)를 각기 따로 준비하고, 각 메모리 뱅크 내에 저장된 텍셀들과 텍스쳐 공간상의 텍셀의 위치가 도 4(각 텍셀영역 내의 숫자는 저장되는 메모리 뱅크를 표시한다)와 같이 맵핑 되게 저장함으로써 동시에 8개의 텍셀을 엑세스해서 트라이리니어 인터폴레이터로 넘기고 이들을 한 클럭사이클 만에 인터폴레이션해서 대표되는 텍셀 값을 구할 수 있는 구조가 제

안되었다.

<31> 그러나 실제 3 차원 그래픽 시스템의 응용에서는 다양한 크기의 텍스쳐 이미지들이 이용되고 있으므로, 도 3과 같은 텍스램(texram) 구조가 그대로 적용되기 위해서는 앞으로 사용되어질 텍스쳐 이미지 보다 충분히 큰 메모리가 준비되어 있지 않으면 안되고, 매 픽셀을 그릴 때마다 LOD 값이 크게 변화하는 경우는 메모리 속의 내용을 계속해서 변경하기 위해 많은 시간이 소모되게 된다. 이는 곧 비효율적인 메모리 사용 및 가격상승의 요인이 되므로 도 3과 같은 구조는 현실적으로 구현해서 사용하기에는 문제가 있다.

<32> 또 하나의 통용되는 개념으로는 도 5와 같이 밉맵 상에서 LOD 레벨이 낮아질수록 텍스쳐 이미지 저장을 위한 공간이 기하급수적으로 늘어나게 되므로, 상위 수 개의 LOD 레벨의 텍스쳐들만 현재 시스템 메모리 위로 전부 올리고(clipmap pyramid), 나머지 LOD 레벨의 값들은 현재 화면상에 그려지는 부분만을 올려두고(clipmap stack) 필요에 따라서 다른 부분들은 전체 밉맵을 저장하고 있는 하드디스크로부터 가져와서 이용함으로써, 대형의 텍스쳐 이미지를 이용해서 텍스쳐 맵핑을 할 때 시스템 메모리 용량을 크게 줄여 줄 수 있는 방법이 제안되었으며, 이를 클립맵(clipmap)이라 한다.

<33> 그러나 이 클립맵 방법은 하드디스크와 시스템 메모리간에 적용되고 있으며, 모든 동작이 소프트웨어적으로 이루어지므로 시스템 메모리의 용량을 줄이는 것이 주목적이지 텍스쳐 맵핑을 가속하기 위한 방법은 아니다.

【발명이 이루고자 하는 기술적 과제】

<34> 상술한 바와 같은 종래 기술의 문제점을 해결하기 위한 본 발명의 목적은 클립맵의 개념을 시스템 메모리와 텍스쳐 맵핑용 캐쉬 메모리와의 관계에 적용시키면서 하드웨어적으로 밉맵핑을 가속화시킬 수 있는 캐쉬 메모리 및 캐쉬 미스 시 발생하는 페널티를 줄일 수 있는 캐쉬 미스 페널티 저감방법을 제공하는데 있다.

【발명의 구성 및 작용】

<35> 상술한 목적을 달성하기 위한 본 발명의 기술적 사상은 트라이리니어 인터폴레이션을 이용한 밉맵핑 가속을 위하여 적당한 크기의 워킹 셋(working set) 만큼의 텍스쳐만을 저장하는 캐쉬 메모리를 만들고, 한 클럭 사이클만에 트라이리니어 인터폴레이션을 위해서 필요한 8개의 텍셀을 모두 엑세스해서 최종 텍셀의 값을 구해 낼 수 있는 특수한 구조를 갖는 캐쉬를 구성함으로써 다양한 크기의 텍스쳐 이미지들에 대해서 효율적으로 적용이 가능하면서 소형, 저가의 시스템에서도 텍스쳐 맵핑 가속이 가능하도록 하는 것이다.

<36> 또한 캐쉬 미스(cache miss) 발생 시에 발생하는 페널티(penalty)를 줄이기 위해서 앞으로 필요하게 될 텍스쳐들을 하드웨어적으로 예측해서 프리페칭(prefetching)함으로써 캐쉬 미스 시 발생하는 페널티를 줄이는 것이다.

<37> 이하, 본 발명의 바람직한 실시예에 대하여 첨부도면을 참조하여 상세하게 설명한다.

<38> 본 발명의 텍스쳐 맵핑용 캐쉬 메모리의 개념은 도 6에 나타낸 바와 같이 밉맵 상

의 최상위 수개(예를 들어, 5개) LOD 레벨의 텍셀들을 모두 저장하는 클립램 피라미드와 나머지 LOD 레벨 중에서 현재 필요로 하는 워킹 셋 만을 저장하는 클립램 스택으로 구성된다.

<39> 클립램 스택은 4개의 LOD 레벨에서의 워킹 셋을 동시에 저장할 수 있으며, 각 LOD 레벨은 16개(4*4)의 서브클립으로 이루어진다. 이 서브클립들은 캐쉬 미스 발생시 뿐만 아니라, 미리 필요한 서브클립을 예측하는 서브클립 예측기(subclip predictor)에 의해서 정상 동작중에도 전체 밉맵을 저장하고 있는 외부 메모리로부터 내용 교체가 가능한 하드웨어적인 예측에 의한 프리페칭이 가능하며, 이로써 캐쉬 미스 페널티를 줄이게 된다.

<40> 상술한 개념의 텍스쳐 맵핑용 캐쉬 메모리(클립램)의 구조는 도 7에 나타낸 바와 같다.

<41> 즉, 최상위 수개(예를 들어, 5개)의 LOD 레벨에 대한 텍셀들을 모두 저장하는 클립램 피라미드를 형성하는 제1 DRAM 뱅크(10)와, 나머지 LOD 레벨에서 현재 필요로 하는 워킹 셋만을 저장하는 클립램 스택을 형성하는 제2 DRAM 뱅크(20)와, 상기 제1, 2 DRAM 뱅크(10)(20)들은 트라이리니어 인터폴레이션을 위한 텍스쳐 읽기와 동시에 외부로부터 새로운 텍스쳐 서브클립을 가져오기 위한 SAM(Serial Access Memory) 포트를 구비하며, 상기 제1, 2 DRAM 뱅크(10)(20)의 SAM 포트에 연결되며 외부 메모리로부터 새로운 텍스쳐 서브클립을 가져오는 서브클립 로더(30)와, 밉맵상의 2개 층에서 각각 4개의 텍셀을 가져와 트라이리니어 인터폴레이션을 수행하는 트라이리니어 인터폴레이터(40) 및 캐쉬 미스시의 페널티를 줄이기 위하여 하드웨어적인 예측에 의하여 서브클립을 프리페칭 하

는 서브클립 예측기(50)와, 상술한 텍스쳐 맵핑용 캐쉬 메모리 구성부 전체를 제어하는 제어기(60) 및 현재 스크린에 그려질 픽셀의 텍스쳐 공간상으로 맵핑된 LOD 및 (u, v) 좌표가 제어기(60)에 입력되면, 이 LOD 및 (u, v) 좌표를 중심으로 정수 좌표 위치에 존재하는 8개의 텍셀이 제1, 2 DRAM 뱅크(10)(20)에 존재하는지를 체크하는 CAM(Content Addressable Memory : 21)로 구성된다.

<42> 상술한 구성을 갖는 텍스쳐 맵핑용 캐쉬 메모리의 기본 동작에 대하여 설명한다.

<43> 먼저, 클립램 피라미드를 위한 제1 DRAM 뱅크(10) 상에는 현재 텍스쳐를 위한 최상위 수개 LOD의 모든 텍셀들이 다 존재하므로 실행 중에 내용의 변경이 필요 없다. 그러나 클립램 스택을 위한 제2 DRAM 뱅크(20) 상에는 현재 필요로 하는 워킹 셋만을 저장하고 있으므로 실행 중에 계속해서 내용의 변경이 필요하다. 현재 스크린에 그려질 픽셀의 텍스쳐 공간상으로 맵핑된 LOD 및 (u, v) 좌표가 제어기(60)에 입력되면, 이 LOD 및 (u, v) 좌표를 중심으로 정수 좌표 위치에 존재하는 8개의 텍셀이 존재하는지를 CAM(Content Addressable Memory : 21)을 이용해서 제1, 2 DRAM 뱅크(10)(20)를 체크하고 존재하면 8개의 텍셀을 동시에 읽어서 트라이리니어 인터폴레이터(40)에서 트라이리니어 인터폴레이션을 행한다.

<44> 만일, 8개의 텍셀이 모두 존재하지 않으면 캐쉬 미스가 발생되고, 이 때 제어기(60)는 서브클립 로더(30)로 하여금 필요로 하는 텍셀들이 들어 있는 서브클립을 외부메모리로부터 가져오게 하고, 트라이리니어 인터폴레이터(40)는 이들을 읽어서 트라이리니어 인터폴레이션을 행하게 된다.

<45> 캐쉬 미스시의 페널티를 줄이기 위하여 본 발명의 텍스쳐 맵핑용 캐쉬 메모리는 서브클립 예측기(50)를 가지고 있으며, 이는 앞으로 곧 필요할 서브클립을 예측해서 정상적인 텍스쳐 캐쉬동작 중에 현재 엑세스되고 있는 서브클립과 충돌이 일어나지 않는 새로운 서브클립을 외부로부터 로드할 수 있다.

<46> 본 발명의 일 실시예에 따른 텍스쳐 맵핑용 캐쉬 메모리의 서브클립 예측 기능을 보다 상세하게 설명하면, 텍스쳐 맵핑용 캐쉬 메모리의 내용은 서브클립 단위로 하드웨어적인 예측에 의한 프리페칭이 가능하며, 하드웨어적으로 필요한 서브클립의 예측은 한 스택층 내에서의 서브클립 예측과 스택층 예측, 두 가지로 이루어진다.

<47> 먼저, 한 스택층 내에서의 서브클립 예측에 대하여 설명한다.

<48> 도 8은 한 스택층 내에서 곧 필요하게 될 서브클립의 예측 및 교체 과정을 나타낸 것으로서, 좌측 상단 그림은 3차원 공간상의 도로가 2차원 스크린 위로 투영된 것을 보여주고 있다. 일반적으로 3차원 공간상의 물체는 도면과 같이 작은 삼각형들로 모델링되어지고, 이러한 삼각형들을 저장하는 데이터 파일은 연속된 삼각형들의 집합이다. 그러므로 도 8에서와 같이 도로가 화면상에 그려질 때에는 연속된 삼각형들 영역이 화살표 방향으로 차례대로 그려진다.

<49> 도 8 좌측 하단의 그림은 도로 위에 입혀 질 텍스쳐 이미지로서 4개의 꼭지점 a, b, c, d가 좌측 상단 그림의 도로 위 네 꼭지점 a, b, c, d로 맵핑 되어진다. 그러므로 스크린 상에 연속된 삼각형들이 화살표 방향으로 차례대로 그려지면, 그에 필요한 텍스

쳐 이미지의 엑세스 패턴도 도 8 좌측 하단의 화살표 방향으로 엑세스되어지는 특성을 가지고 있다.

<50> 이러한 특성을 이용해서, 도 8 우측 그림은 특정 스택층 위에 있는 서브클립들이 예측 및 교체되는 것을 보여 주고 있다. 먼저 진하게 칠해진 4*4 서브클립 들(커런트 클립)이 현재 클립램 스택 상에 있는 서브클립들을 나타낸다. (u, v) 트레이스는 텍셀들이 시간에 따라 엑세스되는 패턴의 한 예를 나타내고 있다. 도면에서 내부 2*2 서브클립 경계는 하드웨어적인 예측 한계로서 (u,v) 트레이스가 이 한계를 넘어가게 되면, 지금 당장은 캐쉬 미스가 나지 않았지만 나중에 일어날 캐쉬 미스 시 페널티를 줄여 주기 위해서 곧 필요로 할 서브클립들을 프리페칭 하게 된다. 현재 도면은 (u, v) 트레이스가 우측에 있는 예측 한계를 넘어가는 경우로서 우측에 있는 연하게 칠해진 4개의 서브클립들을 4*4 서브클립들의 좌측 4개의 서브클립이 들어 있는 영역으로 프리페칭 하게 된다. 이로써, 텍스쳐 엑세스 패턴을 이용한 하드웨어적으로 예측된 프리페칭이 가능하며 최상의 경우 캐쉬 미스 없이 언제나 필요한 서브클립들을 미리 텍스쳐 맵핑용 캐쉬 메모리 속에 존재하게 할 수도 있다.

<51> 도 9는 LOD 변화에 따른 필요한 스택층의 예측 및 교체 과정을 나타낸 것으로서, 좌측 상하의 그림은 도 8과 마찬가지이다. 도면에 나타난 것처럼 화살표 방향으로 연속된 삼각형이 그려질 때 LOD의 변화 역시 미소한 변화는 있지만 전체적으로 볼 때 연속적으로 증가하게 된다. 도 9 우측의 그림은 이렇게 연속적으로 증가하는 LOD 트레이스를 보여주고 있다. 커런트 클립램 스택은 현재 클립램 스택 속에 저장된 LOD 레벨을 나타내고 있으며, 내부의 두 개 LOD 레벨이 스택층 예측 한계로 이용된다. 이 예의 경우 LOD

트레이스가 LOD i+2를 넘어서는 순간 LOD i+4에 해당하는 스택층을 LOD i를 저장하고 있던 영역에 로딩함으로써 스택층도 서브클립과 마찬가지로 하드웨어적으로 예측된 프리페칭을 하게 된다.

【발명의 효과】

<52> 상술한 바와 같은 본 발명의 3차원 그래픽 텍스쳐 맵핑용 캐쉬 메모리는 다양한 크기의 텍스쳐 이미지들에 대해서도 트라이리니어 인터폴레이션에 의한 밉맵핑의 하드웨어적인 가속화를 현실적으로 가능하게 하며, 소형·저가의 3차원 그래픽 시스템에 대해서도 적용 가능함으로써 보다 현실감 있고 빠른 3차원 그래픽을 가능하도록 한다.

<53> 본 발명은 특정의 실시예와 관련하여 도시 및 설명하였지만, 특허청구범위에 의해 나타난 발명의 사상 및 영역으로부터 벗어나지 않는 한도 내에서 다양한 개조 및 변화가 가능하다는 것을 당업계에서 통상의 지식을 가진 자라면 누구나 쉽게 알 수 있을 것이다.

【특허청구범위】

【청구항 1】

트라이리니어 인터폴레이션을 위한 텍스쳐 읽기와 동시에 외부로부터 새로운 텍스쳐 서브클립을 가져오기 위한 SAM 포트를 구비한 제1, 2 DRAM 뱅크(10)(20)와;

상기 제1, 2 DRAM 뱅크(10)(20)의 SAM 포트에 연결되며 외부 메모리로부터 새로운 텍스쳐 서브클립을 가져오는 서브클립 로더(30)와;

밉맵상의 2개 층에서 각각 4개의 텍셀을 가져와 트라이리니어 인터폴레이션을 수행하는 트라이리니어 인터폴레이터(40)와;

상기 구성부 전체를 제어하는 제어기(60); 및

현재 스크린에 그려질 픽셀의 텍스쳐 공간상으로 맵핑된 LOD 및 (u, v) 좌표가 제어기(60)에 입력되면, 이 LOD 및 (u, v) 좌표를 중심으로 정수 좌표 위치에 존재하는 8개의 텍셀이 제1, 2 DRAM 뱅크(10)(20)에 존재하는지를 체크하는 CAM(21)을 포함하는 것을 특징으로 하는 3차원 그래픽 텍스쳐 맵핑용 캐쉬 메모리.

【청구항 2】

제1항에 있어서,

상기 제1 DRAM 뱅크(10)는 전체 밉맵 중 수개의 최상위 LOD 레벨의 텍셀 모두를 저장하는 클립램 피라미드를 형성하고, 상기 제2 DRAM 뱅크(20)는 상기 제1 DRAM 뱅크(10)에 저장되지 않은 나머지 LOD 레벨에서 현재 필요로 하는 워킹 셋만을 저장하는 클립램

스택을 형성하는 것을 특징으로 하는 3차원 그래픽 텍스쳐 맵핑용 캐쉬 메모리.

【청구항 3】

제1항에 있어서,

상기 제2 DRAM 뱅크(20)는 밉맵상의 한 LOD 레벨 위에서의 텍스쳐 영역을 n*n 서브 클립들로 나누고, 이들을 단위로 캐쉬 메모리의 내용을 교체하는 것을 특징으로 하는 3 차원 그래픽 텍스쳐 맵핑용 캐쉬 메모리.

【청구항 4】

제1항에 있어서,

한 클럭 사이클만에 트라이리니어 인터폴레이션을 수행할 수 있도록 현재 스크린에 그려질 픽셀의 텍스쳐 공간상으로 맵핑된 LOD 및 (u, v) 좌표를 중심으로 정수 좌표 위치에 존재하는 8개의 텍셀 엑세스를 위한 데이터 패스를 상기 제1, 2 DRAM 뱅크(10)(20)와 트라이리니어 인터폴레이터(40) 사이에 구비하는 것을 특징으로 하는 3차원 그래픽 텍스쳐 맵핑용 캐쉬 메모리.

【청구항 5】

제1항 내지 제4항 중 어느 한 항에 있어서,

캐쉬 미스시의 페널티를 줄이기 위하여 앞으로 곧 필요할 서브클립을 하드웨어적인 예측에 의하여 프리페칭 하는 서브클립 예측기(50)를 더 포함하는 것을 특

징으로 하는 3차원 그래픽 텍스쳐 맵핑용 캐쉬 메모리.

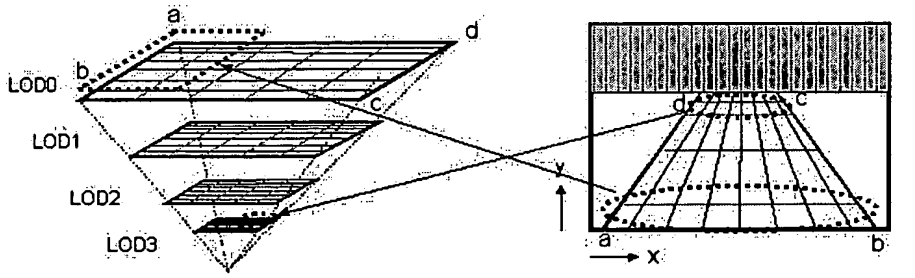
【청구항 6】

커런트 클립램 스택상에 있는 서브클립(4*4) 내부의 2*2 서브클립 경계를 하드웨어적인 서브클립 예측 한계로 하여, (u, v) 트레이스가 이 한계를 넘어가게 되면 한계를 넘어가는 방향(좌·우·상·하)의 서브클립을 프리페칭 하는 한 스택층 내에서의 서브클립 예측과;

커런트 클립램 스택은 현재 클립램 스택 속에 저장된 LOD 레벨(LOD i ~ LOD i+3)을 나타내며 내부의 두 개 LOD 레벨을 스택층 예측 한계로 이용하여, LOD 트레이스가 한계를 넘어가는 순간 다음 LOD 레벨에 해당하는 스택층을 프리페칭 하는 스택층 예측을 수행함으로써 캐쉬 미스 시에 발생하는 페널티를 줄이는 것을 특징으로 하는 캐쉬 미스 페널티 저감방법.

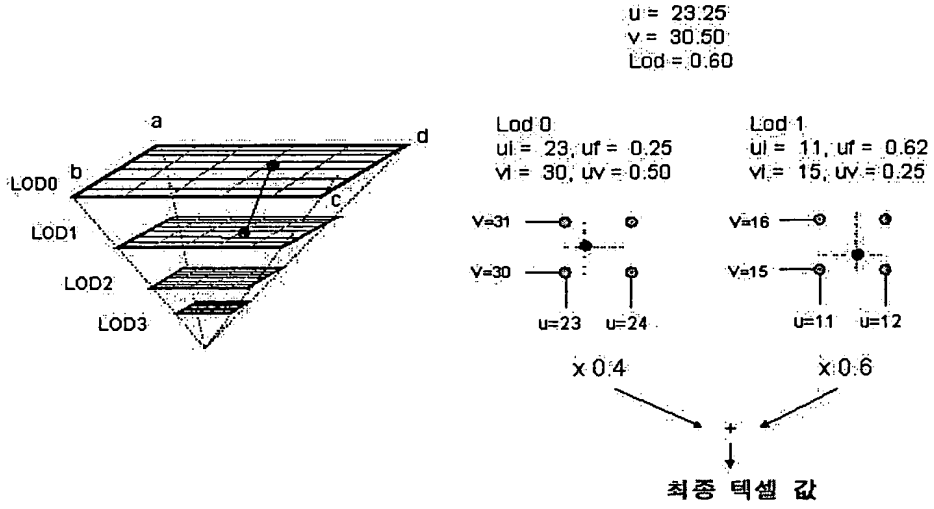
【도면】

【도 1】



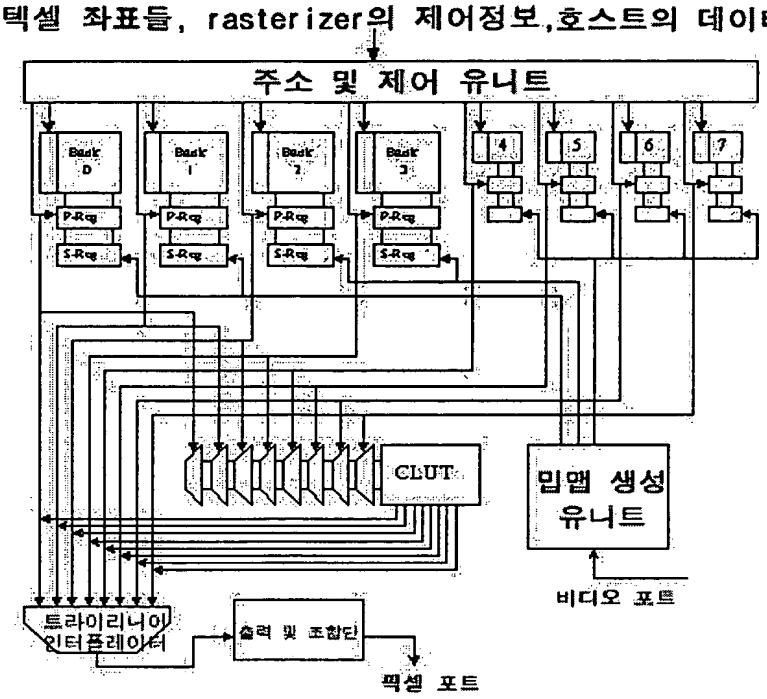
스크린에 투영된 이미지 (2D)

【도 2】



최종 텍셀 값

【도 3】



【도 4】

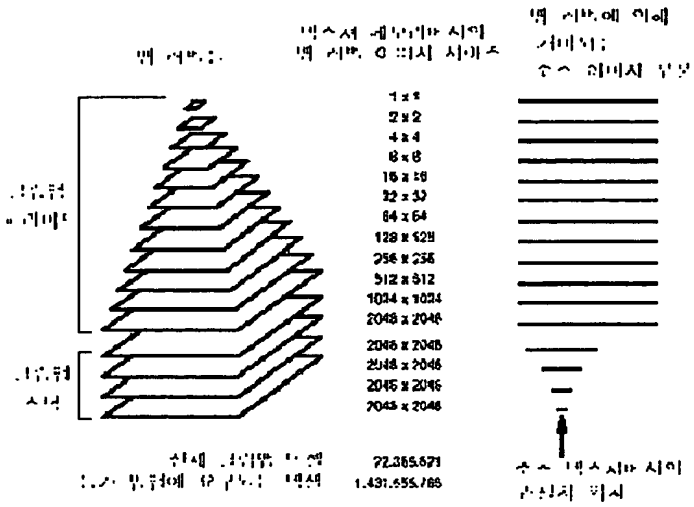
0	1	0	1	0	1	0	1	0	1	0	1
2	3	2	3	2	3	2	3	2	3	2	3
0	1	0	1	0	1	0	1	0	1	0	1
2	3	2	3	2	3	2	3	2	3	2	3
0	1	0	1	0	1	0	1	0	1	0	1
2	3	2	3	2	3	2	3	2	3	2	3
0	1	0	1	0	1	0	1	0	1	0	1
2	3	2	3	2	3	2	3	2	3	2	3
0	1	0	1	0	1	0	1	0	1	0	1
2	3	2	3	2	3	2	3	2	3	2	3
0	1	0	1	0	1	0	1	0	1	0	1
2	3	2	3	2	3	2	3	2	3	2	3

(LOD i)

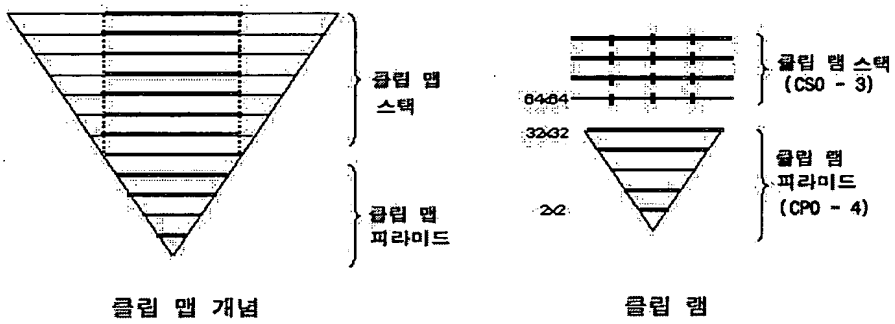
4	5	4	5	4	5
6	7	6	7	6	7
4	5	4	5	4	5
6	7	6	7	6	7
4	5	4	5	4	5
6	7	6	7	6	7

(LOD i+1)

【도 5】

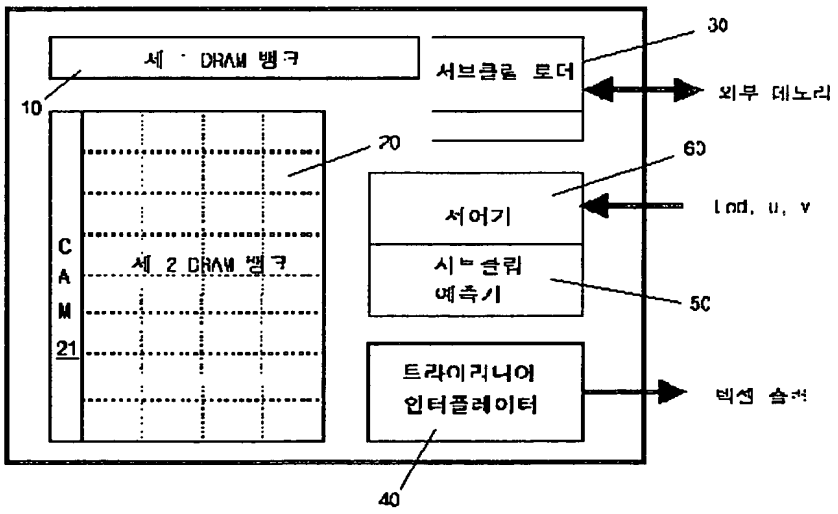


【도 6】

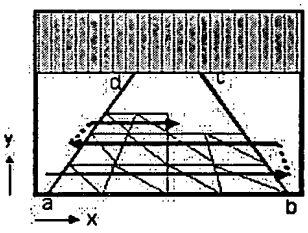


클립 맵 개념 클립 램

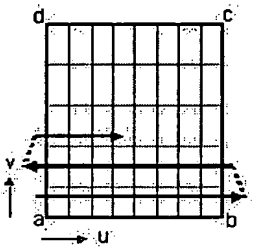
【도 7】



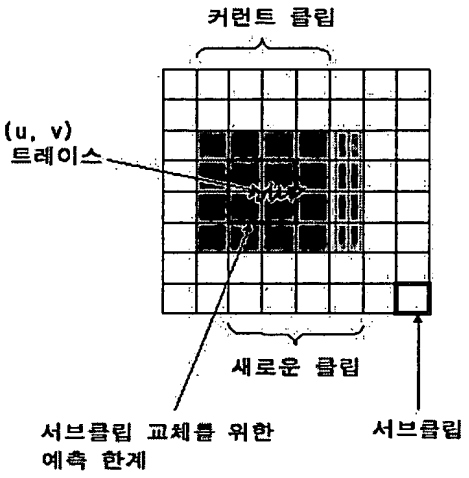
【도 8】



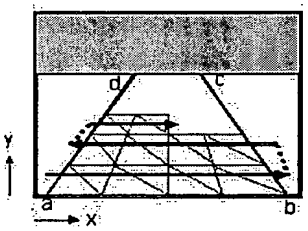
스크린에 투영된 도로 이미지(2D)



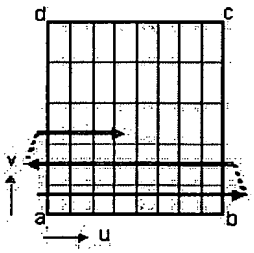
도로 텍스쳐(2D)



【도 9】



스크린에 투영된 도로 이미지(2D)



도로 텍스쳐(2D)